

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matic Stare

**Napovedovanje možnih trkov med  
vlaki in predori**

MAGISTRSKO DELO

MAGISTRSKI ŠTUDIJSKI PROGRAM DRUGE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA  
SMER: RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Uroš Čibej

Ljubljana, 2025



To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 4.0 Mednarodna* (CC BY-SA 4.0). To pomeni, da se lahko besedilo, slike, grafi in druge sestavine tega dela prosto delijo, reproducirajo, priobčujejo javnosti in predelujejo za vsak namen, tudi komercialno, če se jasno navede avtor (po možnosti tudi naslov in povezava do izvirnika), priloži povezava do licence, označijo morebitne spremembe, pri predelavah pa se uporabi enaka licenca (CC BY-SA 4.0). Licenca ne dovoljuje dodajanja dodatnih pravnih ali tehničnih omejitev in ne velja za dele, za katere nosilec pravic ni avtor. Podrobnosti licence so dostopne na spletni strani <https://creativecommons.org>.





## ZAHVALA

*Zahvaljujem se svojemu mentorju doc. dr. Uroš Čibej za strokovno pomoč, spodbudo in potrpežljivost pri nastajanju tega magistrskega dela. Hvala tudi vsem ostalim, ki so mi vsa leta pomagali pri študiju in mi stali ob strani.*

*Matic Stare, 2025*



Vsem rožicam tega sveta.

*"The only reason for time is so that everything doesn't happen at once."*

— Albert Einstein



# Kazalo

## Povzetek

## Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Opis problema . . . . .	1
1.2	Motivacija in cilji dela . . . . .	1
1.3	Prispevki magistrske naloge . . . . .	2
1.4	Struktura magistrske naloge . . . . .	3
<b>2</b>	<b>Pregled sorodnih del</b>	<b>5</b>
2.1	Zaznavanje trkov v prostoru . . . . .	5
2.2	Obdelava oblakov točk . . . . .	6
2.3	Metode prostorskega indeksiranja . . . . .	6
2.4	Analitično modeliranje v železniškem prometu . . . . .	6
2.5	Prehodne krivulje v železniškem prometu . . . . .	7
2.6	Primerjava obstoječih pristopov . . . . .	7
<b>3</b>	<b>Teoretične osnove</b>	<b>9</b>
3.1	Geometrijska predstavitev predorov . . . . .	9
3.2	Geometrijski model vagona . . . . .	10
3.2.1	Pozicioniranje z medosno razdaljo . . . . .	11
3.2.2	Ortogonalni koordinatni sistem . . . . .	11
3.2.3	Simulacija premika vagona . . . . .	11

## KAZALO

3.2.4	Kritične točke za računanje trkov . . . . .	12
3.3	B-zlepki (B-splines) . . . . .	13
3.3.1	Matematične osnove B-zlepkov . . . . .	13
3.3.2	Interpolacija in aproksimacija . . . . .	14
3.4	Metode za merjenje razdalj in določanje strani . . . . .	14
3.4.1	Razdalja do parametrične krivulje . . . . .	14
3.4.2	Določanje strani . . . . .	15
3.5	Prileganje tovora v prilagojen model vagona . . . . .	15
<b>4</b>	<b>Metodologija in pristop</b>	<b>17</b>
4.1	Pregled predlaganega pristopa . . . . .	17
4.2	Predobdelava vhodnih podatkov . . . . .	18
4.2.1	Transformacija v 3D koordinatni sistem . . . . .	18
4.2.2	Generiranje horizontalnih prerezov . . . . .	18
4.2.3	Klasifikacija točk na levo in desno steno . . . . .	18
4.3	Strategija zaznavanja trkov . . . . .	19
4.3.1	Izbira kritičnih točk . . . . .	19
4.3.2	Protokol preverjanja kršitev . . . . .	19
4.4	Analitične metode za optimizacijo prileganja . . . . .	20
<b>5</b>	<b>Implementacija</b>	<b>21</b>
5.1	Arhitektura sistema . . . . .	21
5.2	Konfiguracija sistema . . . . .	22
5.2.1	Parametri predorov . . . . .	22
5.2.2	Parametri vagona . . . . .	22
5.2.3	Parametri simulacije . . . . .	22
5.2.4	Parametri vizualizacije . . . . .	23
5.3	Ključni programski moduli . . . . .	23
5.3.1	ExcelParser – predobdelava podatkov . . . . .	23
5.3.2	TunnelSlicer – obdelava geometrije . . . . .	24
5.3.3	Wagon – modeliranje vagona . . . . .	25
5.3.4	CollisionDetector – zaznavanje trkov . . . . .	25

## *KAZALO*

5.3.5	CollisionMarginsToMesh – generiranje prilagojenega modela vagona . . . . .	26
5.3.6	FitCargo – prileganje tovora . . . . .	26
5.3.7	Simulation – koordinacija sistema . . . . .	26
5.3.8	Main – glavna vstopna točka sistema . . . . .	27
5.4	Numerične metode in optimizacija . . . . .	27
5.4.1	Izvedba B-zlepkov s SciPy . . . . .	27
5.4.2	Numerično iskanje ničel za presečišča . . . . .	27
5.5	Vizualizacija s PyVista . . . . .	28



# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>2D</b>	two-dimensional	dvodimenzionalno
<b>3D</b>	three-dimensional	tridimenzionalno
<b>B-zlepek</b>	B-spline	B-zlepek
<b>JSON</b>	JavaScript Object Notation	JSON
<b>LiDAR</b>	light detection and ranging	svetlobna detekcija in razdalje
<b>MP4</b>	MPEG-4 video	MPEG-4 video
<b>STL</b>	stereolithography	stereolitografija
<b>VTK</b>	visualization toolkit	vizualizacijski komplet



# Povzetek

**Naslov:** Napovedovanje možnih trkov med vlaki in predori

V vzorcu je predstavljen postopek priprave magistrskega dela z uporabo okolja L<sup>A</sup>T<sub>E</sub>X. Vaš povzetek mora sicer vsebovati približno 100 besed, ta tukaj je odločno prekratek. Dober povzetek vključuje: (1) kratek opis obravnavnega problema, (2) kratek opis vašega pristopa za reševanje tega problema in (3) (najbolj uspešen) rezultat ali prispevek magistrske naloge.

## Ključne besede

*računalniška geometrija, železniški promet, zaznavanje trkov, oblaki točk, B-zlepki, brušenje vagona, 3D vizualizacija, optimizacija*



# Abstract

**Title:** Predicting Possible Collisions Between Trains and Tunnels

This sample document presents an approach to typesetting your BSc thesis using L<sup>A</sup>T<sub>E</sub>X. A proper abstract should contain around 100 words which makes this one way too short. A good abstract contains: (1) a short description of the tackled problem, (2) a short description of your approach to solving the problem, and (3) (the most successful) result or contribution in your thesis.

## Keywords

*computer geometry, rail transport, collision detection, point clouds, B-splines, wagon shaving off, 3D visualization, optimization*



# Poglavlje 1

## Uvod

### 1.1 Opis problema

V železniškem prometu je zagotavljanje varnosti v predorih ključnega pomena, še posebej pri dolgi in široki tovorni kompoziciji, ki se giblje skozi ozke in ukrivljene predore. Problem, ki ga obravnavam v tej magistrski nalogi, je zaznavanje morebitnih trkov med vlakom in stenami predora, ki nastanejo zaradi nepravilnega sledenja predpisankemu varnostnemu prostoru ali napak v modeliranju geometrije predora.

Klasične metode, kot je uporaba minimalnega prereza, so v takšnih scenarijih nezadostne, saj ne upoštevajo kompleksne ukrivljenosti poti ali relativnih premikov vagona, ki lahko presežejo varnostne meje, zlasti v ostrih ovinkih. Problem je izrazit pri dolgi tovorni kompoziciji, kjer razlika med položajem sprednje in zadnje osi povečuje tveganje za trk. Poleg tega trenutne metode pogosto niso dovolj prilagodljive za različne geometrije predorov in vlakov.

### 1.2 Motivacija in cilji dela

Motivacija za delo izhaja iz realnega izziva, ki sem ga prejel od podjetja Slovenske železnice. Ti so izrazili potrebo po razvoju avtomatiziranega sistema,

ki bi omogočil natančno zaznavanje trkov med vlakom in predorom.

V tej nalogi predlagam pristop, ki temelji na obdelavi oblaka točk predora in analitičnem modeliranju gibanja vlaka. Osnovna vhodna podatka sta oblak točk predora, pridobljen s 3D laserskim skenerjem, in kontrolne točke, ki definirajo pot železniške proge. Na podlagi teh podatkov sistem obdela geometrijo predora, generira B-zlepke za stene predora v različnih horizontalnih plasteh ter simulira gibanje vagona vzdolž kontrolnih točk. Med simulacijo se izvaja zaznavanje trkov s preverjanjem razdalj med kritičnimi točkami vagona in stenami predora. Na podlagi tega se generira največji doposten model vagona, ki še omogoča varen prehod skozi predor. Na koncu v ta model vključimo tudi prileganje tovora različnih oblik.

Naloga se umešča na področje računalniškega modeliranja in analize v prostoru ter prinaša novost v kombinaciji obdelave oblakov točk s simulacijo gibanja vlaka in zaznavanjem trkov v realnem času.

### 1.3 Prispevki magistrske naloge

Magistrska naloga bo prispevala k razvoju sistema za zaznavanje trkov med vlakom in predorom s simulacijo gibanja vlaka. V primerjavi z obstoječimi metodami, ki temeljijo na statični analizi minimalnih prerezov, predlagana rešitev omogoča dinamično simulacijo gibanja in kontinuirano preverjanje varnostnih razdalj.

Novost naloge je v integraciji obdelave oblakov točk predora z analitičnim modeliranjem gibanja vagona vzdolž ukrivljene poti ter implementaciji sistema za zaznavanje trkov v realnem času. Glavni prispevki magistrske naloge so:

- Razvoj sistema za obdelavo oblakov točk predora z B-zlepki za reprezentacijo sten
- Implementacija simulacije gibanja vagona vzdolž kontrolnih točk z ortogonalnim koordinatnim sistemom

- Sistem za zaznavanje trkov z analizo razdalj med kritičnimi točkami vagona in stenami predora
- Generiranje največjega dopustnega modela vagona za varen prehod skozi predor
- Algoritem prileganja tovora različnih oblik v prilagojen model vagona
- Praktična aplikacija za Slovenske železnice z možnostjo nadaljnjega razvoja

## 1.4 Struktura magistrske naloge

Magistrska naloga je organizirana v sedem poglavij. Po uvodu v Poglavlju 1 sledi pregled sorodnih del v Poglavlju 2. Teoretične osnove so predstavljene v Poglavlju 3, metodologija v Poglavlju 4. Implementacija je opisana v Poglavlju 5, eksperimentalno ovrednotenje v Poglavlju ???. Sklepne ugotovitve so podane v Poglavlju ??.



# Poglavlje 2

## Pregled sorodnih del

### 2.1 Zaznavanje trkov v prostoru

Na področju zaznavanja trkov v prostoru se pogosto uporabljajo metode, ki temeljijo na analizi oblakov točk in algoritmih prostorskega indeksiranja. Ena izmed najpogosteje uporabljenih tehnik je uporaba k-d dreves za učinkovito iskanje sosednjih točk v prostoru, kot je prikazano v delu Schauerja in Nüchterja [1]. Prednost njihovega pristopa je visoka računska učinkovitost pri analizi oblakov točk velikega obsega. Ker pa je točk zelo veliko, se poraja potreba po bolj pametnih izračunih trkov. Njihov članek bo služil kot osnova za to magistrsko delo.

Kot alternativo klasičnim metodam so Hermann et al. [2] razvili algoritme, ki temeljijo na vokselizaciji prostora. Ti algoritmi omogočajo hitro preverjanje prostorske zasedenosti, vendar lahko pri zelo natančnih analizah izgubijo detajle zaradi diskretizacije prostora.

V delu Niwa in Masuda [3] je predstavljen pristop za zaznavanje trkov z metodo globinskih slik, kar izboljša učinkovitost in pravilnost. Ta pristop omogoča zanesljivejše zaznavanje trkov v gostih oblakih točk, vendar ima še vedno veliko časovno in prostorsko zahtevnost.

## 2.2 Obdelava oblakov točk

Klein in Zachmann [4] obravnavata zaznavanje trkov s pomočjo implicitnih površin, ustvarjenih iz oblakov točk. Njihov pristop je posebej uporaben pri obdelavi kompleksnih geometrij, vendar je računsko zahteven, kar lahko omejuje uporabo v realnem času.

Avtorji Li et al. [5] pregledajo najnovejše pristope strojnega učenja za obdelavo LiDAR podatkov. Poudarjajo, kako lahko globoko učenje izboljša zaznavanje in analizo oblakov točk v avtonomnih vozilih, še posebej pri neenakomernih in šumnih podatkih. Kljub napredku se metode soočajo z izzivi pri obdelavi velikih oblakov točk in zagotavljanju rezultatov v realnem času, kar omejuje njihovo uporabnost v hitro spremenjajočih se okoljih.

## 2.3 Metode prostorskega indeksiranja

Prostorsko indeksiranje je ključno za učinkovito obdelavo velikih oblakov točk. K-d drevesa, kot jih uporablja Schauer in Nüchter [1], omogočajo hitro iskanje najbližjih sosedov v večdimenzionalnih prostorih. Te strukture podatkov so posebej primerne za aplikacije, kjer je potrebno pogosto iskanje točk v določeni okolini.

Vendar pa tradicionalne metode prostorskega indeksiranja pogosto niso optimalne za dinamične scenarije, kjer se objekti gibljejo skozi prostor. V takšnih primerih je potreben pristop, ki upošteva časovno komponento gibanja.

## 2.4 Analitično modeliranje v železniškem prometu

Everett et al. [6] predstavijo sistem za izogibanje trkom v dinamičnih okoljih z uporabo globokega spodbujevalnega učenja. Prednost tega pristopa je prilagodljivost za različne scenarije in obdelava spremenljivega števila agen-

tov brez strogih predpostavk o njihovem gibanju. Kljub temu metoda manj poudarja analizo geometrijskih lastnosti, kar jo omejuje pri natančnih prostorskih analizah, kot je analiza trkov med vlakom in predorom, zaradi česar je njena uporaba v tem kontekstu manj primerna.

## 2.5 Prehodne krivulje v železniškem prometu

V železniškem prometu so prehodne krivulje ključne za zagotavljanje gladkega prehoda med ravnimi in ukrivljenimi odseki prog. Brustad in Dalmo [7] analizirajo prehodne krivulje, ki omogočajo gladek prehod med ravnimi in ukrivljenimi odseki železniških tirov. Glavna prednost teh krivulj je njihova sposobnost zmanjšanja sil in obrabe vozil ter tirnic, kar povečuje udobje potnikov in zmanjšuje stroške vzdrževanja. Kljub temu se raziskave na tem področju še vedno soočajo z izzivi, kot so določanje optimalnih lastnosti krivulj za različne scenarije in vozne profile.

Jiang et al. [8] predlagajo uporabo paraboličnih in sinusoidnih prehodnih krivulj za zmanjšanje dolgovalovnih nepravilnosti v vertikalnih profilih tirov. Prednost tega pristopa je zmanjšanje pospeškov pri prehodih, kar izboljša stabilnost vlaka in varnost potnikov. Slabost pa je, da metoda zahteva natančno načrtovanje in prilagoditev specifičnim konstrukcijskim zahtevam, kar lahko poveča začetne stroške implementacije.

## 2.6 Primerjava obstoječih pristopov

Iz zgoraj predstavljenih del je razvidno, da večina obstoječih metod zanemarja dinamične lastnosti gibanja ali ne omogoča učinkovitega prilagajanja različnim geometrijam. Metode, ki temeljijo na obdelavi oblakov točk [1, 3], so računsko zahtevne in pogosto niso primerne za analizo v realnem času. Po drugi strani pristopi strojnega učenja [5, 6] omogočajo prilagodljivost, vendar ne zagotavljajo teoretično podprtih rezultatov, ki so potrebni za varnostno kritične aplikacije v železniškem prometu.

Cilj te magistrske naloge je preseči omejitve obstoječih pristopov z vključitvijo analitičnega modeliranja gibanja kritičnih točk in prekrivanjem teh krivulj z geometrijo predora, kar bo omogočilo natančnejše in hitrejše zaznavanje trkov.

# Poglavlje 3

## Teoretične osnove

V tem poglavju predstavimo teoretične osnove, ki so potrebne za razumevanje predlaganega pristopa k zaznavanju trkov med vlakom in predorom. Osredotočimo se na geometrijsko predstavitev objektov v prostoru, matematične osnove B-zlepakov ter metode za merjenje razdalj in določanje strani. Poglavlje zaključimo z matematičnim opisom postopka prileganja tovora v prilagojen model vagona.

### 3.1 Geometrijska predstavitev predorov

Predori so predstavljeni z oblaki točk, pridobljenih s 3D laserskim skenerjem. Oblak točk predstavlja diskretno vzorčenje površine predora, kjer vsaka točka  $\mathbf{p}_i = (x_i, y_i, z_i)$  določa prostorsko pozicijo na steni predora. Oblak točk  $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$  vsebuje  $n$  točk, ki skupaj opisujejo geometrijo predora.

V našem pristopu obravnavamo podatke predorov, ki so organizirani kot zaporedni prečni prerezi vzdolž osi predora. Vsak presek je definiran z množico 2D koordinat  $(x, y)$ , ki opisujejo obris predora na določeni poziciji vzdolž osi  $z$ .

Glavni izliv pri obdelavi takšnih podatkov je njihova transformacija v uporaben 3D oblak točk, ki ohranja geometrijske lastnosti predora. To dosežemo z naslednjim pristopom:

- **Postavitev oblaka točk v 3D:** 2D koordinate  $(x, y)$  za vsak presek dopolnimo z  $z$  koordinato, ki ustreza poziciji vzdolž poti
- **Transformacija koordinatnega sistema:** podatki se transformirajo iz globalnega koordinatnega sistema v lokalni koordinatni sistem predora, kjer  $z$ -os sovpada s središčno linijo predora

Matematična formulacija transformacije je podana s:

$$\mathbf{p}'_i = \mathbf{R}(\mathbf{p}_i - \mathbf{t}_{\text{center}}) \quad (3.1)$$

, kjer je  $\mathbf{R}$  rotacijska matrika, izpeljana iz tangentnega vektorja središčne linije,  $\mathbf{t}_{\text{center}}$  pa translacijski vektor do središča predora.

Za konstrukcijo rotacijske matrike  $\mathbf{R}$  uporabimo Rodriguesovo rotacijsko formulo [9]:

$$\mathbf{R} = \mathbf{I} + \sin(\theta)\mathbf{K} + (1 - \cos(\theta))\mathbf{K}^2 \quad (3.2)$$

, kjer je  $\theta$  kot med vektorjema,  $\mathbf{K}$  pa antisimetrična matrika osi rotacije. Namen te transformacije je prilagoditi oblak točk dejanskemu poteku predora v prostoru. Ker so prvotni podatki organizirani kot ravni prerezi, rotacijska transformacija zagotovi, da se oblak točk čim bolje prilega dejanski ukrivljeni geometriji predora. To je ključno za natančno predstavitev prostorskih odnosov med vlakom in stenami predora ter posledično za zanesljivo zaznavanje trkov.

## 3.2 Geometrijski model vagona

Vagon je modeliran kot pravokotno telo (kvader) z dimenzijami širina  $w$ , višina  $h$  in globina  $d$ . Geometrijska predstavitev temelji na konceptu medosne razdalje in ortogonalnem koordinatnem sistemu.

### 3.2.1 Pozicioniranje z medosno razdaljo

Pozicija vagona je določena s točkama  $\mathbf{p}_0$  in  $\mathbf{p}_1$ , ki sta oddaljeni za medosno razdaljo  $l_{wb}$ . Za določitev sprednje osi uporabljamo metodo presečišča kroga in krivulje.

Sprednja os  $\mathbf{p}_1$  se nahaja na presečišču kroga s središčem v  $\mathbf{p}_0$  in radijem  $l_{wb}$  z B-zlepkom kontrolnih točk:

$$\|\mathbf{C}(t) - \mathbf{p}_0\|^2 = l_{wb}^2 \quad (3.3)$$

, kjer je  $\mathbf{C}(t)$  parametrična predstavitev B-zlepka kontrolnih točk in  $t \in [0, 1]$  parameter krivulje.

Za rešitev te enačbe iščemo ničle funkcije:

$$f(t) = \|\mathbf{C}(t) - \mathbf{p}_0\|^2 - l_{wb}^2 \quad (3.4)$$

Ta pristop zagotavlja, da je medosna razdalja natančno ohranjena tudi na ukrivljenih odsekih poti, kar je ključno za realistično modeliranje gibanja vagona.

### 3.2.2 Ortogonalni koordinatni sistem

Orientacija vagona v prostoru je določena z ortogonalnim koordinatnim sistemom:

$$\mathbf{f} = \frac{\mathbf{p}_1 - \mathbf{p}_0}{\|\mathbf{p}_1 - \mathbf{p}_0\|} \quad (\text{naprej}) \quad (3.5)$$

$$\mathbf{u} = (0, 1, 0) \quad (\text{navzgor}) \quad (3.6)$$

$$\mathbf{r} = \mathbf{u} \times \mathbf{f} \quad (\text{desno}) \quad (3.7)$$

### 3.2.3 Simulacija premika vagona

Za potrebe simulacije gibanja vagona je treba določiti celoten geometrijski model, ki omogoča natančno predstavitev vagona v prostoru. Vagon je mo-

deliran z osmimi vogali pravokotnega telesa, ki se izračunajo na podlagi dejanskih mej vagona.

Najprej definirajmo sprednjo in zadnjo mejo vagona:

$$\mathbf{p}_{\text{rear}} = \mathbf{p}_0 - \mathbf{f} \cdot (d \cdot \omega) \quad (3.8)$$

$$\mathbf{p}_{\text{front}} = \mathbf{p}_0 + \mathbf{f} \cdot (d \cdot (1 - \omega)) \quad (3.9)$$

, kjer je  $\omega$  parameter odmika koles,  $d$  globina vagona in  $\mathbf{f}$  enotski vektor v smeri naprej.

Osem vogalov vagona je nato definiranih kot:

$$\mathbf{v}_{i,j,k} = \mathbf{p}_{\text{base}} + i \cdot \frac{w}{2} \mathbf{r} + j \cdot h \cdot \mathbf{u} + k \cdot \frac{d}{2} \cdot \mathbf{f} \quad (3.10)$$

, kjer so  $i, k \in \{-1, 1\}$ ,  $j \in \{0, 1\}$  in  $\mathbf{p}_{\text{base}} = \frac{\mathbf{p}_{\text{rear}} + \mathbf{p}_{\text{front}}}{2}$  je središče med zadnjo in sprednjo mejo vagona.

Ta pristop omogoča popolno geometrijsko predstavitev vagona v prostoru ter je ključen za vizualizacijo in analizo gibanja celotne strukture vagona vzdolž ukrivljene poti.

### 3.2.4 Kritične točke za računanje trkov

Za učinkovito zaznavanje trkov se na vsaki višinski ravnini  $y$  v predoru uporablja optimiziran nabor šestih kritičnih točk, ki predstavljajo najkritičnejše pozicije za možne kolizije s stenami. Te točke so razporejene v treh vzdolžnih pozicijah (zadaj, sredina, spredaj) na obeh stranskih robovih vagona.

Kritične točke za zaznavanje kolizij na višini  $y$  so definirane s formulo:

$$\mathbf{c}_{s,p} = \mathbf{p}_{\text{base},p} + s \cdot \frac{w}{2} \mathbf{r} + y \cdot \mathbf{u} \quad (3.11)$$

, kjer je:

$$s \in \{-1, 1\} \quad (\text{levo/desno})$$

$$p \in \{\text{back, middle, front}\} \quad (\text{zadaj, sredina, spredaj})$$

$$\mathbf{p}_{base,back} = \mathbf{p}_{\text{rear}}$$

$$\mathbf{p}_{base,middle} = \frac{\mathbf{p}_{\text{rear}} + \mathbf{p}_{\text{front}}}{2}$$

$$\mathbf{p}_{base,front} = \mathbf{p}_{\text{front}}$$

, kjer je  $\mathbf{r}$  enotski vektor v desno smer,  $\mathbf{u}$  enotski vektor navzgor in  $w$  širina vagona.

Ta pristop generira za vsako horizontalno ravnino predora (definirano z  $y$  koordinato) natanko šest kritičnih točk, ki pokrivajo celotno širino in dolžino vagona. Sistematična razporeditev omogoča zanesljivo zaznavanje kršitev varnostnih razdalj ali položajev, kjer bi vagon bil zunaj dovoljenih mej predora.

### 3.3 B-zlepki (B-splines)

B-zlepki so parametrične krivulje, ki omogočajo gladko interpolacijo ali aproksimacijo množice točk. V svojem sistemu jih uporabljam za reprezentacijo sten predora ter za modeliranje poti vagona.

#### 3.3.1 Matematične osnove B-zlepkov

B-zlepek stopnje  $p$  [10] je definiran s kontrolnimi točkami  $\mathbf{C}_0, \mathbf{C}_1, \dots, \mathbf{C}_n$  in vozliščnim vektorjem  $\mathbf{t} = \{t_0, t_1, \dots, t_{n+p+1}\}$ :

$$\mathbf{S}(y) = \sum_{i=0}^n \mathbf{C}_i B_{i,p}(y) \tag{3.12}$$

, kjer so  $B_{i,p}(y)$  B-zlepek bazne funkcije, definirane rekurzivno:

$$B_{i,0}(y) = \begin{cases} 1 & \text{če } t_i \leq y < t_{i+1} \\ 0 & \text{sicer} \end{cases} \quad (3.13)$$

$$B_{i,p}(y) = \frac{y - t_i}{t_{i+p} - t_i} B_{i,p-1}(y) + \frac{t_{i+p+1} - y}{t_{i+p+1} - t_{i+1}} B_{i+1,p-1}(y). \quad (3.14)$$

### 3.3.2 Interpolacija in aproksimacija

Pri obdelavi oblakov točk predora uporabljamo dva pristopa:

**Interpolacija:** B-zlepek natančno prehaja skozi vse podane točke. To je ustrezno, ko želimo ohraniti vse detajle geometrije predora.

**Aproksimacija:** B-zlepek se prilagodi splošni obliki podatkov, vendar ne prehaja natančno skozi vse točke. To je koristno za zmanjšanje vpliva šuma v meritvah.

## 3.4 Metode za merjenje razdalj in določanje strani

Za zaznavanje trkov je ključno merjenje razdalj med točkami vagona in stenami predora ter določanje, na kateri strani B-zlepka se točka nahaja.

### 3.4.1 Razdalja do parametrične krivulje

Razdalja med točko  $\mathbf{q}$  in B-zlepkom  $\mathbf{C}(u)$  je definirana kot:

$$d(\mathbf{q}, \mathbf{C}) = \min_{u \in [0,1]} \|\mathbf{q} - \mathbf{C}(u)\| \quad (3.15)$$

To je optimizacijski problem, ki ga rešujemo numerično z iskanjem najbližje točke na krivulji.

### 3.4.2 Določanje strani

Za določitev, ali točka je znotraj ali zunaj predora, uporabljamo predznak razdalje. Za 2D preseku predora uporabimo vektorski produkt za določitev strani:

$$s = \text{sign}(\mathbf{t} \times (\mathbf{q} - \mathbf{C}(u^*))) \quad (3.16)$$

, kjer pozitivni predznak pomeni, da je točka na levi strani krivulje, negativni pa na desni.

## 3.5 Prileganje tovora v prilagojen model vagona

Prileganje tovora v prilagojen model vagona je geometrijski problem [11], kjer je cilj najti takšno postavitev tovora, da v celoti je znotraj omejitev vagona. Tovor je predstavljen kot 3D objekt poljubne oblike, prilagojen model vagona pa 3D objekt, ki je rezultat analize prostorskih omejitev v predoru.

Matematično je postopek prileganja definiran kot iskanje transformacije, ki vključuje rotacijo in translacijo tovora, tako da je za vsako točko tovora izpolnjen pogoj vsebovanosti v prostoru vagona:

$$\forall \mathbf{p}_i \in \mathcal{C} : \quad T(\mathbf{p}_i) \in \mathcal{V} \quad (3.17)$$

, kjer je  $\mathcal{C}$  množica točk tovora,  $\mathcal{V}$  prostornina prilagojenega vagona,  $T$  pa transformacija, ki vključuje rotacijo in translacijo.

Rotacija je opisana z Eulerjevimi koti  $(\alpha, \beta, \gamma)$ , translacija pa z vektorjem  $(t_x, t_y, t_z)$ . Transformacija  $T$  je sestavljena iz zaporedja rotacij in premikov:

$$T(\mathbf{p}) = R(\alpha, \beta, \gamma) \cdot \mathbf{p} + \mathbf{t} \quad (3.18)$$

, kjer je  $R$  rotacijska matrika,  $\mathbf{t}$  translacijski vektor.

Iskanje ustrezne transformacije je diskretizirano po rotacijskih in translacijskih parametrih. Za vsako kombinacijo se preveri, ali je pogoj vsebovanosti

izpolnjen za vse točke tovora. Če obstaja vsaj ena kombinacija, kjer je pogoj izpolnjen, je prileganje možno.

Pri kompleksnih oblikah tovora in vagona se za preverjanje vsebovanosti uporablja prostorsko vzorčenje oziroma pretvorba objektov v oblake točk. S tem se problem reducira na množico geometrijskih testov med točkami tovora in prostornino vagona.

Ta pristop omogoča analizo prileganja poljubnih oblik tovora v poljubno prilagojen model vagona, ne glede na kompleksnost geometrije. Rezultat je množica transformacijskih parametrov, ki omogočajo varen prevoz tovora skozi predor.

# Poglavlje 4

## Metodologija in pristop

### 4.1 Pregled predlaganega pristopa

Predlagani pristop k zaznavanju trkov med vlakom in predorom temelji na kombinaciji obdelave oblakov točk ter analitičnega modeliranja gibanja vagona. Sistem je zasnovan kot večstopenjski proces, ki omogoča natančno in učinkovito zaznavanje potencialnih kršitev varnostnih razdalj.

Pristop se začne z obdelavo vhodnih podatkov, kjer transformiramo 2D prereze predora v 3D koordinatni sistem ter jih organiziramo v horizontalne plasti. Vsaka plast se nato obdela s pomočjo B-zlepkov, ki omogočajo gladko reprezentacijo sten predora. Vzporedno z obdelavo predora se izvaja modeliranje gibanja vagona vzdolž kontrolnih točk z uporabo koncepta medosne razdalje.

Ključna novost pristopa je v dinamičnem preverjanju razdalj med kritičnimi točkami vagona in stenami predora med simulacijo gibanja. Za razliko od statičnih metod, ki analizirajo le minimalne prereze, naš pristop upošteva dejanski geometrijski model vagona in njegovo orientacijo v prostoru z uporabo šestih kritičnih točk, ki so definirane z enačbo (3.11).

## 4.2 Predobdelava vhodnih podatkov

### 4.2.1 Transformacija v 3D koordinatni sistem

Vhodni podatki predora so organizirani kot zaporedni prečni prerezi vzdolž njegove osi, kjer vsak presek opisuje 2D obris predora na določeni poziciji.

Njihova transformacija v 3D koordinatni sistem temelji na teoriji transformacij iz poglavja 3.1, specifično na enačbi (3.1).

Za vsako pozicijo vzdolž poti se določi lokalni koordinatni sistem na osnovi tangentnega vektorja. Uporabi se Rodriguesova rotacijska formula za poravnavo lokalnega sistema z ukrivljeno potjo, pri čemer se zagotavlja, da transformacija ne popači oblike prerezov in omogoča gladek prehod med sosednjimi prerezi.

### 4.2.2 Generiranje horizontalnih prerezov

Po transformaciji v 3D se oblak točk organizira v horizontalne prerezne na različnih višinskih nivojih. Ta pristop omogoča obravnavo kompleksne 3D geometrije kot zaporedja 2D problemov, zmanjša računsko kompleksnost in omogoča vzporedno obdelavo različnih višinskih nivojev. Skrbno se določijo višinski nivoji ter tolerance za vključevanje točk v posamezne prerezne.

### 4.2.3 Klasifikacija točk na levo in desno steno

Za vsak horizontalni presek se točke sistemsko klasificirajo na levo in desno steno predora. Pristop temelji na geometrijski analizi prostorskih odnosov z uporabo konceptov iz teorije določanja strani (3.16).

Določi se središčna linija prerezna kot referenčna os, nato se z vektorskim produktom določi stran za vsako točko. Zagotavlja se enotna orientacija preko vseh prerezov in preverja logičnost klasifikacije.

## 4.3 Strategija zaznavanja trkov

### 4.3.1 Izbira kritičnih točk

Strategija zaznavanja trkov temelji na konceptu kritičnih točk, ki predstavljajo najverjetnejše lokacije za nastanek kolizij. Pristop upošteva geometrijske lastnosti vagona za identifikacijo robnih točk z najvišjim tveganjem za trk ter dinamiko gibanja, kjer se različni deli vagona gibljejo po različnih poteh. Hkrati se išče optimalen kompromis med natančnostjo in računsko učinkovitostjo.

Izbere se šest kritičnih točk (3.11) na vsaki višini. Te točke pokrivajo tri vzdolžne pozicije (zadaj, sredina, spredaj) na obeh stranskih robovih vagona in zagotavljajo reprezentativnost za celotno strukturo vagona.

### 4.3.2 Protokol preverjanja kršitev

Protokol preverjanja kršitev definira sistematičen pristop k zaznavanju situacij, kjer vagon presega dovoljene varnostne meje.

V vsaki poziciji vagona se izračunavajo pozicije kritičnih točk z uporabo ortogonalnega koordinatnega sistema (3.7). Za vsako kritično točko se določijo najbližje pozicije na stenah predora ter izračunajo razdalje in orientacija glede na stene z metodami iz poglavja 3.4. Izračunane razdalje se sistematično primerjajo z varnostnimi mejami, zaznane kršitve pa se beležijo in kategorizirajo.

Sistem razlikuje med dvema kategorijama kršitev: situacijami, kjer je del vagona zunaj fizičnih mej predora, in kršitvami varnostne razdalje, kjer je razdalja do stene manjša od predpisane varnostne meje.

Ta pristop omogoča zgodnje opozarjanje pred potencialnimi trki in je posebej koristen pri dolgih kompozicijah, kjer se lahko kršitve razvijajo postopoma preko več pozicij. Celoten sistem je zasnovan za kontinuirano delovanje med simulacijo gibanja, kar omogoča analizo varnosti v realnem času.

## 4.4 Analitične metode za optimizacijo prileganja

Analiza prileganja se začne z iskanjem največjega možnega vagona za vsak predor. Postopek temelji na sistematičnem preverjanju razdalj med kritičnimi točkami vagona in stenami predora, pri čemer se varnostna razdalja začasno izključi. S tem se določi prostorska omejitev, ki jo predor dejansko dopušča.

Na podlagi teh omejitev se generira prilagojen 3D model vagona, ki izkorišča celoten razpoložljiv prostor. Model je rezultat interpolacije izmerjenih razdalj na površini vagona in upošteva vse relevantne geometrijske omejitve.

Za tovore nepravilnih oblik se prileganje preverja z diskretnim vzorčenjem različnih orientacij in položajev tovora v prilagojenem vagonu. Vsaka kombinacija rotacije in translacije se preveri glede na pogoj vsebovanosti. Če je pogoj izpolnjen, je prevoz možen.

Celoten postopek je sestavljen iz treh korakov: določitev omejitev, generiranje prilagojenega modela in preverjanje prileganja tovora. Rezultati vključujejo parametre postavitve tovora in vizualizacijo ustreznih rešitev.

# Poglavlje 5

## Implementacija

V tem poglavju predstavljamo tehnično realizacijo sistema za zaznavanje trkov med vlakom in predorom. Implementacija sledi metodološkemu pristopu iz poglavja 4 in je zasnovana kot modularen sistem z osmimi ključnimi komponentami.

### 5.1 Arhitektura sistema

Sistem je implementiran v programskem jeziku Python z uporabo knjižnic NumPy, SciPy, pandas in PyVista za vizualizacijo. Arhitektura sledi modularnemu pristopu, kjer vsak modul opravlja specifično nalogu v procesu zaznavanja trkov.

Glavni tok izvajanja se začne z branjem Excelovih datotek, ki vsebujejo podatke o prerezih predora. Ti se transformirajo v 3D oblak točk z uporabo kontrolnih točk, ki definirajo pot železniške proge. Geometrija predora se organizira v horizontalne plasti, kjer se za vsako plast generirajo B-zlepki sten. Vzporedno se simulira gibanje vagona vzdolž kontrolnih točk, pri čemer se kontinuirano preverjajo razdalje med kritičnimi točkami vagona in stenami predora.

## 5.2 Konfiguracija sistema

Sistem omogoča prilagajanje parametrov preko konfiguracijskih slovarjev, kar zagotavlja fleksibilnost pri testiranju različnih scenarijev in prilagajaju različnim tipom predorov ter vlakov. Konfiguracija je organizirana v štiri glavne kategorije parametrov.

### 5.2.1 Parametri predorov

Geometrijski model predora se konfigurira z naslednjimi parametri:

- `control_points_offset`: vertikalni odmik središča tirnic
- `tunnel_center_offset`: vertikalni odmik središča predora
- `train_max_height`: maksimalna višina vlaka

### 5.2.2 Parametri vagona

Geometrijski model vagona se konfigurira z naslednjimi parametri:

- `train_width`: širina vagona (privzeto 3200 mm)
- `train_height`: višina vagona (privzeto 3900 mm)
- `train_depth`: globina vagona (privzeto 6000 mm)
- `wheel_offset`: odmik osi koles (privzeto 0.25, kar pomeni 25% globine vagona)

### 5.2.3 Parametri simulacije

Simulacija se nadzoruje z naslednjimi parametri:

- `n_horizontal_slices`: število horizontalnih plasti za analizo geometrie (privzeto 30)

- `wall_spline_degree`: stopnja B-zlepkov za stene predora (privzeto 3)
- `safety_margin`: varnostna razdalja v milimetrih (privzeto 300 mm)
- `stop_on_safetyViolation`: prekinitve simulacije ob kršitvi (privzeto `true`)
- `export_mp4`: izvoz animacije v MP4 format (privzeto `false`)

#### 5.2.4 Parametri vizualizacije

Sistem omogoča prilagoditev vizualizacije preko PyVista knjižnice:

- Barve objektov (vagon: modra, kršitve: oranžna/rumena)
- Velikosti točk in črt za prikaz oblaka točk ter B-zlepkov
- Nastavitev kamere in osvetlitve za optimalno vizualizacijo

Ta modularna konfiguracija omogoča hitro prilagajanje sistema različnim testnim scenarijem brez potrebe po sprememjanju osnovne kode. Preverjanje skladnosti parametrov (npr. `assert train_height <= train_max_height`) zagotavlja, da konfiguracija ne vodi v neveljavne simulacije.

### 5.3 Ključni programske moduli

#### 5.3.1 ExcelParser – predobdelava podatkov

Modul `excel_parser.py` implementira funkcionalnosti za branje in transformacijo vhodnih podatkov. Glavni funkciji `parse_excel_to_points_dict()` in `efficient_data_loading()` omogočata pretvorbo Excel datotek v strukturirane podatke za nadaljnjo obdelavo.

Funkcija `parse_excel_to_points_dict()` bere Excel datoteko in organizira podatke v slovar, kjer ključi predstavljajo razdalje vzdolž poti, vrednosti pa koordinate X, Y, Z za vsak presek. 2D koordinate se dopolnijo

z Z koordinato, izračunano iz pozicije vzdolž poti, pomnožene s faktorjem `space_out_factor` za normalizacijo razdalj.

Za optimizacijo hitrosti nalaganja se implementira `efficient_data_loading()`, ki uporablja Parquet format za shranjevanje obdelanih podatkov. Ob prvem branju se Excel datoteka prevede in shrani v Parquet format, pri nadalnjih uporabah pa se podatki naložijo neposredno iz Parquet datoteke, kar znatno skrajša čas nalaganja.

Dodatno se implementira funkcionalnost `prepare_control_points()`, ki zagotavlja generiranje ali branje kontrolnih točk iz datoteke. Če kontrolne točke obstajajo v predpomnilni datoteki, se naložijo iz `control_points.txt`, sicer se generirajo iz podatkov predora.

### 5.3.2 TunnelSlicer – obdelava geometrije

Razred `TunnelSlicer` implementira kompleksno obdelavo geometrije predora. Njegova glavna naloga je transformacija 2D prerezov predora v 3D oblak točk ter generiranje horizontalnih plasti z B-zlepki sten.

Ključna metoda `_curve_points()` implementira transformacijo koordinatnega sistema z uporabo Rodriguesove rotacijske formule iz enačbe (3.1). Za vsak presek se določi lokalni koordinatni sistem na osnovi tangentnega vektorja kontrolnih točk ter izvede rotacijska transformacija, ki poravna presek z ukrivljeno potjo predora.

Metoda `_generate_splines_at_y_values()` organizira oblak točk v horizontalne prereze na različnih višinskih nivojih. Za vsak nivo se točke klasificirajo na levo in desno steno z uporabo funkcije `_classify_based_on_b_spline()`, ki implementira teorijo določanja strani iz enačbe (3.16). Iz klasificiranih točk se nato generirajo B-zlepki z uporabo SciPy funkcij `splprep()` in `splev()`.

Za učinkovitost se implementira predpomnenje transformiranih podatkov v Parquet format. Metodi `save_to_parquet()` in `load_from_parquet()` omogočata shranjevanje in branje obdelanega oblaka točk, kar prepreči ponovne izračune pri večkratnih zaganjanjih sistema.

### 5.3.3 Wagon – modeliranje vagona

Razred `Wagon` implementira geometrijski model vagona in njegovo pozicioniranje vzdolž ukrivljene poti. Vagon je modeliran kot pravokotno telo z osmimi vogali, definiranim z dimenzijami širina, višina in globina.

Metoda `_calculate_orthogonal_coordinate_system()` implementira pozicioniranje vagona z uporabo koncepta medosne razdalje (3.3). Za določitev sprednje osi se uporabi metoda `_find_point_at_distance()`, ki rešuje presečišče kroga in krivulje z numeričnim iskanjem ničel funkcije s pomočjo enačbe (3.4).

Izvedba uporablja SciPy funkciji `root_scalar()` z metodo `bracket` za zanesljivo iskanje presečišča ter `minimize_scalar()` v primerih, ko presečišče ni najdeno. Ta pristop zagotavlja, da je medosna razdalja natančno ohranjena tudi na ostrih ovinkih.

Metoda `get_vertices_at_position()` izračuna vseh osem vogalov vagona z uporabo ortogonalnega koordinatnega sistema (3.7). Vogali se določijo na osnovi dejanskih mej vagona, ki upoštevajo parameter `wheel_offset` za pozicioniranje osi koles.

### 5.3.4 CollisionDetector – zaznavanje trkov

Razred `CollisionDetector` implementira sistem za zaznavanje kršitev varnostnih razdalj med vagnom in stenami predora. Uporablja strategijo kritičnih točk (3.11).

Metoda `get_wagon_points()` generira šest kritičnih točk vagona na določeni višini: tri vzdolžne pozicije (zadaj, sredina, spredaj) na obeh stranskih robovih. Te točke predstavljajo najverjetnejše lokacije za nastanek kolizij.

Glavna metoda `check_collision()` izvaja sistematično preverjanje vseh kritičnih točk za vse horizontalne plasti predora. Za vsako kritično točko se določi najbližja pozicija na ustrezni steni z metodo `find_closest_point_on_curve()`, ki implementira iskanje najbližje točke na B-zlepku ter določanje strani z vektorskim produktom.

Sistem razlikuje med dvema tipoma kršitev: situacijami, kjer se del va-

gona nahaja zunaj fizičnih mej predora (`outside_tunnel`), in krštvami varnostne razdalje (`too_close`), kjer je razdalja do stene manjša od predpisane varnostne meje.

### 5.3.5 CollisionMarginsToMesh – generiranje prilagojenega modela vagona

Modul `collision_margins_to_mesh.py` pretvori izmerjene varnostne razdalje iz JSON datoteke v 3D mrežni model. Postopek ustvari plasti točk na različnih višinah, kjer je vsaka plast sestavljena iz 6 točk (tri vzdolžne pozicije na obeh straneh vagona). Te točke se premikajo za izmerjene razdalje: desne točke premakne v levo, leve točke v desno.

Mrežni model se konstruira s povezavanjem sosednjih plasti z bočnimi ploskvami ter zaključka s spodno in zgornjo ploskvijo. Rezultat se shrani v format VTK.

### 5.3.6 FitCargo – prileganje tovora

Modul `fit_cargo.py` preveri, ali se tovor poljubne oblike prilega v prilagojen model vagona. To doseže z diskretnim vzorčenjem različnih orientacij in položajev tovora. Funkcija `mesh_to_points()` pretvori 3D mrežo tovora v oblak točk z adaptivno diskretizacijo. Funkcija `can_fit()` preveri, ali so vse točke tovora vsebovane v modelu vagonu pri dani orientaciji (yaw, pitch, roll).

Funkcija `find_any_fit()` sistematično preizkuša različne orientacije s prilagajalno stopnjo diskretizacije (privzeto  $90^\circ$ ). Za nepravilne oblike tovora omogoči hitro iskanje prve veljavne orientacije.

### 5.3.7 Simulation – koordinacija sistema

Razred `Simulation` služi kot glavna povezovalna komponenta, ki koordinira delovanje vseh višjih modulov. Implementira glavni simulacijski cikel, upravljanje kamere ter izvoz rezultatov.

### 5.3.8 Main – glavna vstopna točka sistema

`main.py` vsebuje konfiguracijski sistem, ki omogoča izbiro med različnimi načini delovanja in predori. Uporabnik lahko preizkusi različne geometrije proge z nastavljivo krivuljnimi funkcijami.

Podprtji so štirje načini simulacije: osnovna analiza s standardnimi dimenzijami vagona, merjenje varnostnih mej z večjim modelom, uporaba prilagojenega modela vagona ter simulacija z realističnim 3D modelom vlaka.

## 5.4 Numerične metode in optimizacija

### 5.4.1 Izvedba B-zlepkov s SciPy

B-zlepki so implementirani z uporabo SciPy knjižnice, specifično funkcij `splprep()` in `splev()`. Funkcija `splprep()` sprejme niz kontrolnih točk in generira parametričen B-zlepek s stopnjo gladkosti `s=0` za interpolacijo skozi vse točke. Parameter `k` določa stopnjo zlepka, privzeto nastavljeno na 3 za kubične zlepke.

Za reprezentacijo sten predora se B-zlepki uporabljajo v 2D prostoru (koordinati X in Z), saj se obdelava izvaja po horizontalnih plasteh. Koordinata Y ostane konstantna za vsako plast. Ta pristop omogoča učinkovito obdelavo kompleksne 3D geometrije kot zaporedja 2D problemov.

### 5.4.2 Numerično iskanje ničel za presečišča

Za določitev položaja vagona vzdolž ukrivljene poti se implementira numerično iskanje ničel z uporabo SciPy funkcije `root_scalar()`. Problem presečišča kroga in krivulje iz enačbe (3.4) se reši z metodo `bracket`, ki zahteva interval z različnim predznakom funkcije na koncih.

Izvedba najprej vzorči funkcijo na 1000 točkah ter išče spremembe predznaka, ki nakazujejo prisotnost ničle. Za vsak tak interval se nato izvede natančno iskanje ničle z metodo `bracket`. V primerih, ko presečišče ni najdeno, se uporabi metoda `minimize_scalar()`, ki poišče najblizjo točko na

krivulji ter aproksimira smer s tangentnim vektorjem.

## 5.5 Vizualizacija s PyVista

Sistem uporablja PyVista knjižnico za 3D vizualizacijo in animacijo simulacije. Izvedba omogoča prikaz oblaka točk predora, B-zlepkov sten, gibanja vagona ter zaznavanja kršitev v realnem času.

Glavna vizualizacija se izvaja preko objekta `pv.Plotter()`, ki omogoča interaktivno 3D sceno. Oblak točk predora se prikaže z metodo `add_points()`, B-zlepki sten pa z `add_mesh()` za polilinijske objekte. Vagon se predstavi kot pravokotno telo z `pv.Cube()`.

Za zaznavanje kršitev se implementira vizualna povratna informacija z barvnimi črtami med kritičnimi točkami vagona in najbližjimi stenami. Oranžne črte označujejo situacije zunaj predora, rumene pa kršitve varnostne razdalje. Dodatno se dodajo besedilne oznake z informacijami o razdalji in tipu kršitve.

Sistem omogoča izvoz animacije v MP4 format z nastavljanjem parametra `export_mp4=True`, kar je koristno za dokumentacijo rezultatov in predstavitev.

# Literatura

- [1] J. Schauer, A. Nüchter, Efficient point cloud collision detection and analysis in a tunnel environment using kinematic laser scanning and k-d tree search, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XL-3 (2014) 289 – 295.  
URL <https://doi.org/10.5194/isprsarchives-XL-3-289-2014>
- [2] A. Hermann, F. Drews, J. Bauer, S. Klemm, A. Roennau, R. Dillmann, Unified gpu voxel collision detection for mobile manipulation planning, *IEEE International Conference on Intelligent Robots and Systems* (09 2014). doi:10.1109/IROS.2014.6943148.
- [3] T. Niwa, H. Masuda, Interactive collision detection for engineering plants based on large-scale point-clouds, *Computer-Aided Design and Applications* 13 (4) (2016) 511–518. arXiv:<https://doi.org/10.1080/16864360.2015.1131546>, doi:10.1080/16864360.2015.1131546.  
URL <https://doi.org/10.1080/16864360.2015.1131546>
- [4] J. Klein, G. Zachmann, Point Cloud Surfaces using Geometric Proximity Graphs, *Computers and Graphics* 28 (6) (2004).
- [5] Y. Li, L. Ma, Z. Zhong, F. Liu, D. Cao, J. Li, M. A. Chapman, Deep learning for lidar point clouds in autonomous driving: A review, arXiv preprint arXiv:2005.09830 (2020). arXiv:2005.09830.  
URL <https://arxiv.org/abs/2005.09830>

- [6] M. Everett, Y. F. Chen, J. P. How, Collision avoidance in pedestrian-rich environments with deep reinforcement learning, *IEEE Access* 9 (2021) 10357–10377. doi:[10.1109/ACCESS.2021.3050338](https://doi.org/10.1109/ACCESS.2021.3050338).  
URL <http://dx.doi.org/10.1109/ACCESS.2021.3050338>
- [7] T. F. Brustad, R. Dalmo, Railway transition curves: A review of the state-of-the-art and future research, *Infrastructures* 5 (5) (2020). doi:[10.3390/infrastructures5050043](https://doi.org/10.3390/infrastructures5050043).  
URL <https://www.mdpi.com/2412-3811/5/5/43>
- [8] L. Jiang, Y. Li, Y. Zhao, M. Cen, The characteristics of long-wave irregularities in high-speed railway vertical curves and method for mitigation, *Sensors* 24 (13) (2024). doi:[10.3390/s24134403](https://doi.org/10.3390/s24134403).  
URL <https://www.mdpi.com/1424-8220/24/13/4403>
- [9] J. S. Dai, Euler-rodrigues formula variations, quaternion conjugation and intrinsic connections, *Mechanism and Machine Theory* 92 (2015) 144–152. doi:<https://doi.org/10.1016/j.mechmachtheory.2015.03.004>.  
URL <https://www.sciencedirect.com/science/article/pii/S0094114X15000415>
- [10] M. S. Hasan, M. N. Alam, M. Fayz-Al-Asad, N. Muhammad, C. Tunç, B-spline curve theory: An overview and applications in real life, *Nonlinear Engineering* 13 (1) (2024) 20240054 [cited 2025-08-16]. doi:[doi:10.1515/nleng-2024-0054](https://doi.org/10.1515/nleng-2024-0054).  
URL <https://doi.org/10.1515/nleng-2024-0054>
- [11] V. Chekanin, Solving the problem of packing objects of complex geometric shape into a container of arbitrary dimension, *Proceedings of the 30th International Conference on Computer Graphics and Machine Vision (GraphiCon 2020). Part 2* (2020) paper50–1doi:[10.51130/graphicon-2020-2-3-50](https://doi.org/10.51130/graphicon-2020-2-3-50).